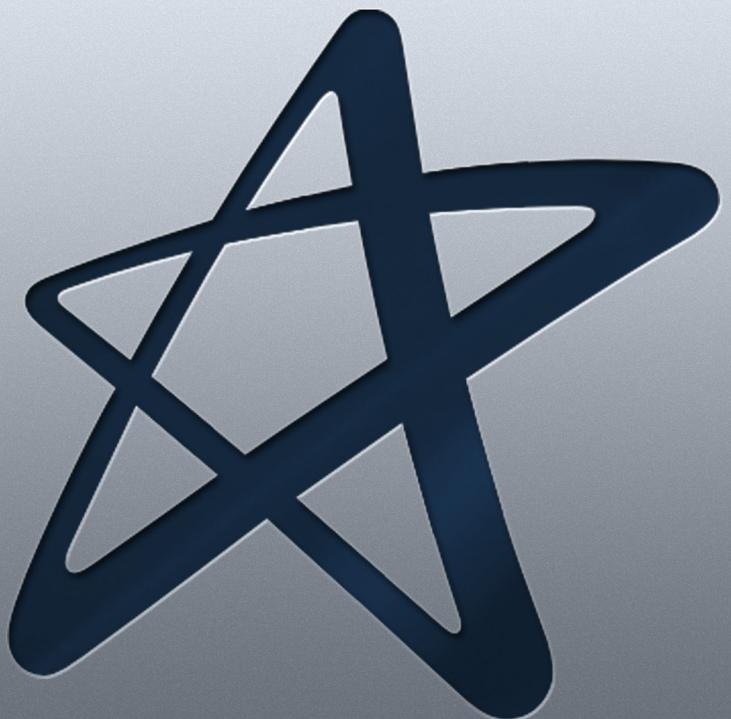


Banco de Dados



Educação a Distância
Cruzeiro do Sul Educacional
Campus Virtual

Material Teórico



Subquerys e Funções de Decisões

Responsável pelo Conteúdo:

Prof. Ms. Alexander Gobbato Albuquerque

Revisão Textual:

Profa. Esp. Vera Lídia de Sá Cicarone

UNIDADE

Subquerys e Funções de Decisões



- Conceito
- Operadores de comparação para múltiplas linhas
- Funções de Decisão



Objetivo de APRENDIZADO

Nessa fase estaremos utilizando vários conceitos aprendidos das outras unidades e acrescentando a esse conhecimento veremos as funções de decisões, essas funções de decisões podem nos auxiliar na formatação dos resultados retornados que irão facilitar em muito o nosso trabalho.

Lembramos a você a importância de realizar todas as atividades propostas dentro do prazo estabelecido para cada unidade. Dessa forma, evitará que o conteúdo se acumule e que você tenha problemas ao final do semestre.

Uma última recomendação: caso tenha problemas para acessar algum item da disciplina ou dúvidas com relação ao conteúdo, não deixe de entrar em contato com seu professor tutor através do botão mensagens.

Contextualização

Com o conhecimento adquirido nos comandos DDL e DML, aprendemos a criar e manipular dados e com esse conhecimento adquirido podemos nos aprofundar um pouco mais nos conceitos de retornos de dados. Veremos nessa unidade que trabalhar com subquery é muito importante porque iremos combinar vários selects para obter um único resultado, esse resultado ainda pode vir acompanhado de uma função de decisão para formatar o resultado ou até mesmo contornar uma possível situação de falha.

Então está na hora de aprendermos a utilizar as subqueries e usufruir das funções de decisões.

Conceito



Subquery é um comando select dentro de um outro comando select que retorna uma ou mais linhas, a fim de satisfazer uma cláusula WHERE.

QUERY PRINCIPAL

SUBQUERY

A subquery (inner query), geralmente, é executada antes da consulta principal.

O resultado da subquery é, então, avaliado pelo da query principal (outer query).

Uma subpesquisa (subquery) é uma declaração SELECT que é aninhada com outra declaração SELECT, a qual retorna resultados intermediários.

A subquery é executada primeiro.

O resultado da subquery é utilizado na query principal.

Estrutura da subquery:

```
SELECT      select_list
FROM        table
WHERE       expr operator
            (SELECT      select_list
             FROM       table);
```

Exemplo:

Quais funcionários possuem um salário maior que o de Abel?

QUERY PRINCIPAL



Qual dos empregados tem salário maior do que o salário de Abel ?

SUBQUERY



Qual o salário de Abel ?

```

SELECT      last_name, salary
FROM        employees
WHERE       salary >
            (SELECT      salary
             FROM        employees
             WHERE       last_name = 'Abel');
    
```

```

SELECT      last_name, salary
FROM        employees
WHERE       salary >
            (SELECT      salary
             FROM        employees
             WHERE       last_name = 'Abel');
    
```

11000 ←

`(SELECT salary
 FROM employees
 WHERE last_name = 'Abel');`

Single Row Subquery
- Retorno **de um único valor** na subquerie

Exemplo:

Mostre o sobrenome (last_name) e o cargo (job_id) dos funcionários que possuem o mesmo cargo do funcionário de número 141.

```

1   SELECT      last_name, job_id
2   FROM        employees
   WHERE       job_id =
            (SELECT      job_id
             FROM        employees
             WHERE       employee_id = 141);
    
```

Exemplo:

Para encontrar todos os empregados que têm o mesmo cargo do David Lee, fazemos:

```

SELECT FIRST_NAME, LAST_NAME, JOB_ID
FROM EMPLOYEES
WHERE JOB_ID = (SELECT JOB_ID
                 FROM EMPLOYEES
                 WHERE FIRST_NAME = 'David' and
                       LAST_NAME = 'Lee')
ORDER BY FIRST_NAME
    
```

`(SELECT job_id
 FROM employees
 WHERE employee_id = 141);`

Exemplo:

Para encontrar todos os empregados do mesmo departamento do David Lee, fazemos:

```
SELECT FIRST_NAME, LAST_NAME, JOB_ID, DEPARTMENT_ID
FROM EMPLOYEES
WHERE DEPARTMENT_ID = (
    SELECT DEPARTMENT_ID
    FROM EMPLOYEES
    WHERE FIRST_NAME = 'David'
    and LAST_NAME = 'Lee')
ORDER BY FIRST_NAME
```

A seguinte pesquisa busca encontrar os empregados que ganham acima da média salarial.

```
SELECT FIRST_NAME, LAST_NAME, JOB_ID,
       DEPARTMENT_ID, SALARY
FROM EMPLOYEES
WHERE SALARY > (SELECT AVG(SALARY)
                  FROM EMPLOYEES)
ORDER BY FIRST_NAME
```

A pesquisa interna retorna o cargo de David Lee (SA REP) o qual é usado na condição WHERE da pesquisa principal.

A seguinte pesquisa busca encontrar os empregados que trabalham nos departamentos de Marketing e Administração.

```
SELECT FIRST_NAME, LAST_NAME, JOB_ID,
       DEPARTMENT_ID, SALARY
FROM EMPLOYEES
WHERE DEPARTMENT_ID IN (SELECT DEPARTMENT_ID
                        FROM DEPARTMENTS
                        WHERE DEPARTMENT_NAME IN
                            ('Marketing', 'Administration'))
ORDER BY FIRST_NAME
```

SingleRowSubquery -
Retorno de **um único valor**
na subquerie

Subpesquisas Aninhadas

Mostrar o primeiro e o último nome, cargo, código do departamento e salário dos empregados cujo salário é maior que o maior salário no departamento ‘SALES’.

```
SELECT FIRST_NAME, LAST_NAME, JOB_ID,
       DEPARTMENT_ID, SALARY
  FROM EMPLOYEES
 WHERE SALARY > ( SELECT MAX(SALARY)
   FROM EMPLOYEES
 WHERE DEPARTMENT_ID = (SELECT DEPARTMENT_ID
   FROM DEPARTMENTS
 WHERE DEPARTMENT_NAME = 'Sales'))
 ORDER BY FIRST_NAME
```

FIRST_NAME	LAST_NAME	JOB_ID	DEPARTMENT_ID	SALARY
Lex	De Haan	AD_VP	90	17000
Neena	Kochhar	AD_VP	90	17000
Steven	King	AD_PRES	90	24000

3 linhas retornadas em 0,00 segundos

[Exportação para CSV](#)

Operadores de comparação para múltiplas linhas



Operator	Significado
IN	Igual a qualquer membro da lista
ANY	Compara o valor com cada valor retornado pela subquerie.
ALL	Compara o valor com todos os valores retornados pela subquerie.
EXISTS	Testa se um valor existe.

Operador ANY

Compara o valor com cada valor retornado pela subquerie.

< ANY = menor que o maior valor

> ANY = maior que o menor valor

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
124	Mourgos	ST_MAN	5800
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

Operador ALL

Compara o valor com todos os valores retornados pela subquerie.

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

Funções de Decisão



DECODE

Ele facilita pesquisas condicionais, fazendo o trabalho de ‘ferramentas’ ou comandos ‘se-então-se não’.

Sintaxe:

DECODE(col/expressão,procurado1,resultado1,...,padrão)

Col/expressão é comparado com cada um dos valores procurados e separados por vírgula e retorna o resultado se a col/expressão for igual ao valor procurado.

Se não for encontrado nenhum dos valores procurados, a função DECODE retorna o valor padrão. Se o valor padrão for omitido, ele retornará um valor nulo.

DECODE - Argumentos

DECODE deve ter, no mínimo, 3 parâmetros ou argumentos.

- ✓ COL/EXPRESSÃO - o nome da coluna ou expressão a ser avaliado.
- ✓ PROCURADO1 - o primeiro valor para ser testado.
- ✓ RESULTADO1- o valor para ser retornado, se o procurado1 for encontrado.
- ✓ PROCURADO1 e RESULTADO1 podem ser repetidos quantas vezes forem necessárias.
(PROCURADO2,RESULTADO2, PROCURADO3,RESULTADO3,...)
- ✓ PADRÃO - o valor a ser retornado, se nenhum procurado for encontrado.

Nota:

col/expressão pode ser vários tipos de dados.

PROCURADO deve ser um dado do tipo coluna ou expressão.

O valor retornado é força para alguns tipos de dados como o terceiro argumento (resultado1).

O seguinte exemplo decodifica os cargos dos tipos IT_PROG e CLERK unicamente. Os outros cargos serão padrão e alterados para INDEFINIDO.

```
SELECT FIRST_NAME, JOB_ID,
DECODE( JOB_ID, 'IT_PROG', 'PROGRAMADOR',
       'FI_ACCOUNT', 'CONTADOR',
       'INDEFINIDO') DECODE_CARGO
FROM EMPLOYEES;
```

FIRST_NAME	JOB_ID	DECODE_CARGO
Steven	AD_PRES	INDEFINIDO
Neena	AD_VP	INDEFINIDO
Lex	AD_VP	INDEFINIDO
Alexander	IT_PROG	PROGRAMADOR
Bruce	IT_PROG	PROGRAMADOR
David	IT_PROG	PROGRAMADOR
Valli	IT_PROG	PROGRAMADOR
Diana	IT_PROG	PROGRAMADOR
Nancy	FI_MGR	INDEFINIDO
Daniel	FI_ACCOUNT	CONTADOR
Há mais de 10 linhas disponíveis. Aumente o seletor de linhas para ver mais linhas.		

10 linhas retornadas em 0,00 segundos

[Exportação para CSV](#)

No exemplo abaixo, nós queremos retornar o salário incrementado de acordo com o tipo de cargo.

```
SELECT FIRST_NAME, JOB_ID, SALARY,
DECODE( JOB_ID, 'IT_PROG', SALARY * 1.1,
       'FI_ACCOUNT', SALARY * 1.2,
       'AD_VP', SALARY * 0.95,
SALARY) DECODE_CARGO
FROM EMPLOYEES;
```

A função condicional case permite o processamento condicional que exija o tratamento de várias hipóteses.

O exemplo seguinte avalia duas expressões lógicas e ainda oferece uma terceira possibilidade, quando as duas anteriores resultarem falsas:

```
SELECT FIRST_NAME, JOB_ID, SALARY,
CASE
    WHEN SALARY < 5000 THEN 'AUMENTO'
    WHEN SALARY > 10000 THEN 'VERIFICAR'
    ELSE 'NÃO AUMENTAR'
END CLASSIFICAÇÃO
FROM EMPLOYEES;
```

Atividades:

- Pesquise na tabela locations (retorne CITY e STATE_PROVINCE), aplicando uma função de decisão no campo State_Province, seguindo a instrução abaixo.

Se STATE_PROVINCE for igual a	Valor Retornado (Criar uma alias para a coluna de retorno Retorno_Decode)
Washington	A String ‘Bem vindo a Casa Branca’
Texas	A String ‘Olá Cowboy ’
California	O campo cidade (CITY)
New Jersey	O campo Street_Address

- Pesquise na tabela employees (retorne o last_name e o salary), usando uma função de decisão no campo Salary, seguindo a instrução abaixo.

Salário	Novo Salário
$\geq 4200 \text{ e } \leq 6000$	Aumento de 10%
$\geq 9000 \text{ e } \leq 17000$	Aumento de 20%
Outros casos	Retorna o salario atual

Material Complementar

Para ajudar a criar toda a estrutura de relacionamento entre as tabelas e gerar os scripts necessários existem esses dois documentos oficiais disponíveis.



- **Erwin** - <http://support.ca.com/cadocs/0/e002953e.pdf>.
- **Dbdesigner** - http://downloads.mysql.com/DBDesigner4/DBDesigner4_manual_1.0.42.pdf.

Referências

FANDERUFF, Damaris. **Dominando o Oracle 9i: Modelagem e desenvolvimento.** São Paulo: Pearson Education do Brasil, 2003.

MORELLI, Eduardo M. Terra, 1996. **Oracle 9i Fundamental: Sql, Pl/SQL e Administração.** São Paulo: Érica, 2002.

Anotações





Educação a Distância
Cruzeiro do Sul Educacional
Campus Virtual

www.cruzeirodosulvirtual.com.br
Campus Liberdade
Rua Galvão Bueno, 868
CEP 01506-000
São Paulo SP Brasil
Tel: (55 11) 3385-3000



Universidade
Cruzeiro do Sul



UNICID
Universidade
Cidade de S. Paulo



UNIFRAN
Universidade
de Franca



UDF
Centro
Universitário



Módulo
Centro
Universitário